

---

# Laboratório de Banco de Dados I

## Parte 07 – Subconsultas

**Prof. Daniel Callegari**  
**Faculdade de Informática – FACIN – PUCRS**

---

### 1. Introdução a Subconsultas

Na linguagem SQL, subconsultas são comandos SELECT aninhados dentro de outros comandos SELECT, INSERT, UPDATE ou DELETE. Podemos ter, inclusive, subconsultas dentro de outras subconsultas. O número de níveis permitido depende do SGBD.

Existem basicamente dois tipos de subconsultas:

- Subconsultas que retornam um único valor; e
- Subconsultas que retornam um conjunto de valores (ou registros).

### 2. Subconsultas que retornam um único valor

Quando uma subconsulta retorna um único valor, usamos os operadores básicos de comparação. Por exemplo, como podemos obter os títulos dos produtos que são mais caros do que o produto cujo código é 142?

Primeiro seria necessário descobrir o preço de tal produto:

```
SELECT preco  
FROM PRODUTOS  
WHERE cod_produto = 142;
```

Esse comando retorna um único valor (221,00). A partir desse resultado, podemos montar o seguinte comando:

```
SELECT titulo  
FROM PRODUTOS  
WHERE preco > 221.00;
```

Mas isso exige duas consultas independentes (e duas trocas de dados entre o aplicativo cliente e o SGBD). Podemos unir os dois comandos SELECT usando o conceito de subconsultas.


Usando uma subconsulta ficaria assim:

```
SELECT titulo
FROM PRODUTOS
WHERE preco >
      (SELECT preco
       FROM PRODUTOS
       WHERE cod_produto = 142);
```

Podemos usar todos os operadores relacionais típicos entre as duas consultas:

>, >=, <, <=, =, <>

**Observação:** Repare que a subconsulta é colocada do lado direito do operador relacional. Coloque a subconsulta entre parênteses.

 Execute os seguintes comandos e identifique que dados eles obtêm.

```
SELECT titulo
FROM PRODUTOS
WHERE importado = 'N' AND preco >
      (SELECT MAX(preco)
       FROM PRODUTOS
       WHERE importado = 'S');
```

```
SELECT ano_lancamento, AVG(preco)
FROM PRODUTOS
GROUP BY ano_lancamento
HAVING AVG(preco) >
      (SELECT AVG(preco)
       FROM PRODUTOS
       WHERE ano_lancamento = trunc(sysdate, 'yyyy'));
```


### 3. Subconsultas que retornam um conjunto de valores

Quando uma subconsulta retornar um conjunto de valores, não podemos usar um operador relacional comum (por quê?). Nestes casos usamos os operadores especiais IN, ANY, SOME ou ALL e EXISTS.

#### 3.1. Subconsultas utilizando o operador IN

Usamos essa construção para obter as linhas iguais a qualquer linha da subconsulta. O operador IN aceita a comparação simultânea de mais de uma coluna. A sintaxe geral é:

```
SELECT nome_da_coluna [...]  
FROM nome_da_tabela [...]  
WHERE [condições AND]  
(nome_da_coluna, nome_da_coluna, ...) [NOT] IN  
      (sub_consulta)
```

 Execute os seguintes comandos e identifique que dados eles obtêm.

```
SELECT PED.num_pedido  
FROM pedidos PED  
WHERE ped.cod_cliente IN  
      (SELECT ADM.cod_administrador  
       FROM administradores ADM);
```

```
SELECT PED.num_pedido  
FROM pedidos PED  
WHERE (PED.cod_cliente, PED.cod_endereco) IN  
      (SELECT END.cod_cliente, END.cod_endereco  
       FROM enderecos END  
       WHERE END.cod_cidade = 20);
```

#### 3.2. Subconsultas utilizando os operadores ANY e SOME

Essa construção compara um valor com cada linha da subconsulta, utilizando um operador de comparação.

Observação: ANY e SOME são sinônimos.

```
SELECT nome_da_coluna [...]  
FROM nome_da_tabela [...]  
WHERE [condições AND]  
nome_da_coluna operador_comparação ANY | SOME  
      (sub_consulta)
```

```
SELECT PROD.titulo
FROM produtos PROD
WHERE PROD.importado = 'N' AND PROD.preco > ANY
      (SELECT PROD1.preco
       FROM produtos PROD1
       WHERE PROD1.importado = 'S');
```

### 3.3. Subconsultas utilizando o operador ALL

Essa construção compara um valor com todas as linhas da subconsulta, utilizando um operador. Por exemplo:

```
SELECT PROD.titulo
FROM produtos PROD
WHERE PROD.importado = 'N' AND PROD.preco > ALL
      (SELECT PROD1.preco
       FROM produtos PROD1
       WHERE PROD1.importado = 'S');
```

## 4. Subconsultas correlacionadas

Uma consulta é dita correlacionada quando ambas as consultas (interna e externa) são interdependentes. Em outras palavras, para cada linha processada da consulta interna, deve-se processar novamente a consulta externa. A consulta interna depende da consulta externa para ser processada.

### 4.1 Subconsultas utilizando o operador EXISTS

Essa construção retorna *verdadeiro* se a subconsulta possuir pelo menos uma linha.

```
SELECT nome_da_coluna [,...]
FROM nome_da_tabela [,...]
WHERE [condições AND]
EXISTS (sub_consulta)
```

Exemplo:

```
SELECT USU.Nome
FROM Usuarios USU, Administradores ADM
WHERE USU.cod_usuario = ADM.cod_administrador AND
      ADM.nivel_privilegio > 2 AND EXISTS
      (SELECT *
       FROM Pedidos PED
       WHERE PED.Cod_Cliente = ADM.cod_administrador)
```

### Dicas finais desta aula

- Coloque as subconsultas entre parênteses.
- As subconsultas devem ficar à direita dos operadores relacionais.
- Não use ORDER BY em uma subconsulta.
- O operador IN é equivalente a “= ANY” e o operador NOT IN, a “!= ALL”.

### Fechamento

Parabéns! Você aprendeu como trabalhar com subconsultas.

1. Alguns tipos de subconsulta substituem um JOIN.
2. Há dados que só podem ser obtidos por meio de subconsultas.

#### Registro de alterações deste documento

Data	Autor	Alterações
26/09/2012	Daniel Callegari	Primeira versão.

-X-